

Шаблон отчёта по лабораторной работе

Простейший вариант

Дмитрий Сергеевич Кулябов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Основы работы с mc	9
4.2	Структура программы на языке ассемблера NASM	12
4.3	Подключение внешнего файла	15
5	Выводы	22
6	Список литературы	23

Список иллюстраций

4.1	Открытый тс	9
4.2	Перемещение между директориями	10
4.3	Создание каталога	11
4.4	Перемещение между директориями	12
4.5	Создание файла	12
4.6	Открытие файла для редактирования	13
4.7	Редактирование файла	13
4.8	Открытие файла для просмотра	14
4.9	Компиляция файла и передача на обработку компоновщику	14
4.10	Исполнение файла	14
4.11	Скачанный файл	15
4.12	Копирование файла	15
4.13	Копирование файла	16
4.14	Редактирование файла	16
4.15	Исполнение файла	17
4.16	Отредактированный файл	17
4.17	Копирование файла	18
4.18	Редактирование файла	19
4.19	Копирование файла	20
4.20	Редактирование файла	21
4.21	Исполнение файла	21

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

1. Основы работы с mc
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

3Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss). Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти: - DB (define byte) — определяет переменную размером в 1 байт; - DW (define word) — определяет переменную размером в 2 байта (слово); - DD (define double word) — определяет переменную размером в 4 байта (двойное слово); - DQ (define quad word) — определяет переменную размером в 8 байт (учетверённое слово); - DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

```
mov dst,src
```

Здесь операнд `dst` — приёмник, а `src` — источник. В качестве операнда могут выступать регистры (`register`), ячейки памяти (`memory`) и непосредственные значения (`const`). Инструкция языка ассемблера `int` предназначена для вызова прерывания с указанным номером.


int `n`

Здесь `n` — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра `sys_calls` `n=80h` (принято задавать в шестнадцатеричной системе счисления).

4 Выполнение лабораторной работы

4.1 Основы работы с mc

Открываю Midnight Commander, введя в терминал mc (рис. 4.1).



```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04/report$ mc
[1]+ Остановлен /usr/bin/mc -P "$MC_PWD_FILE" "$@"
```

Рис. 4.1: Открытый mc

Перехожу в каталог ~/work/study/2022-2023/Архитектура Компьютера/arch-рс, используя файловый менеджер mc (рис. 4.2)

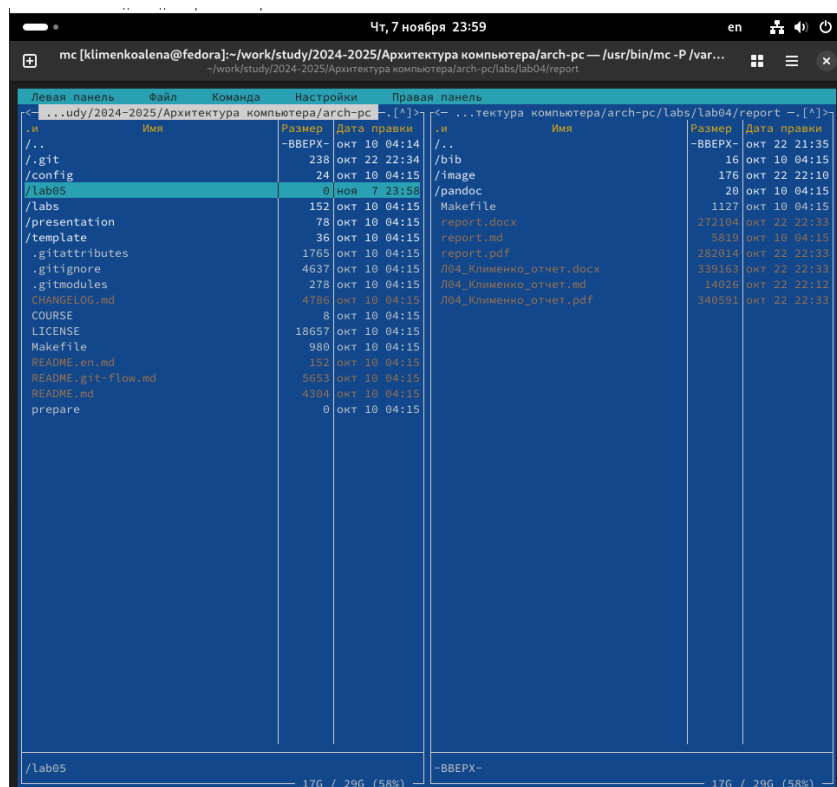


Рис. 4.2: Перемещение между директориями

С помощью функциональной клавиши F7 создаю каталог lab05 (рис. 4.3).

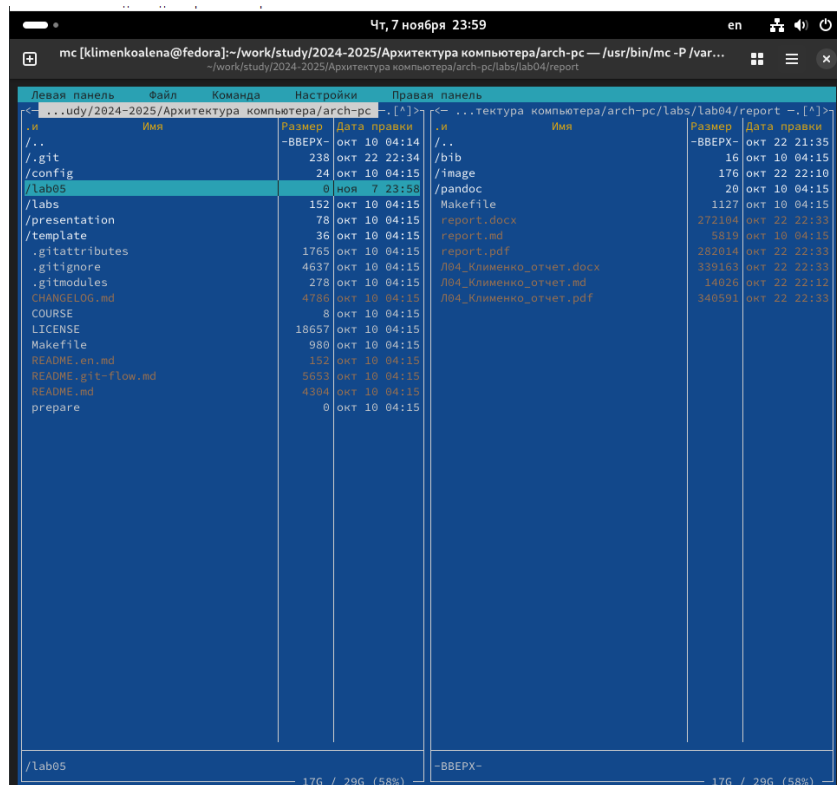


Рис. 4.3: Создание каталога

Переходу в созданный каталог (рис. 4.4).

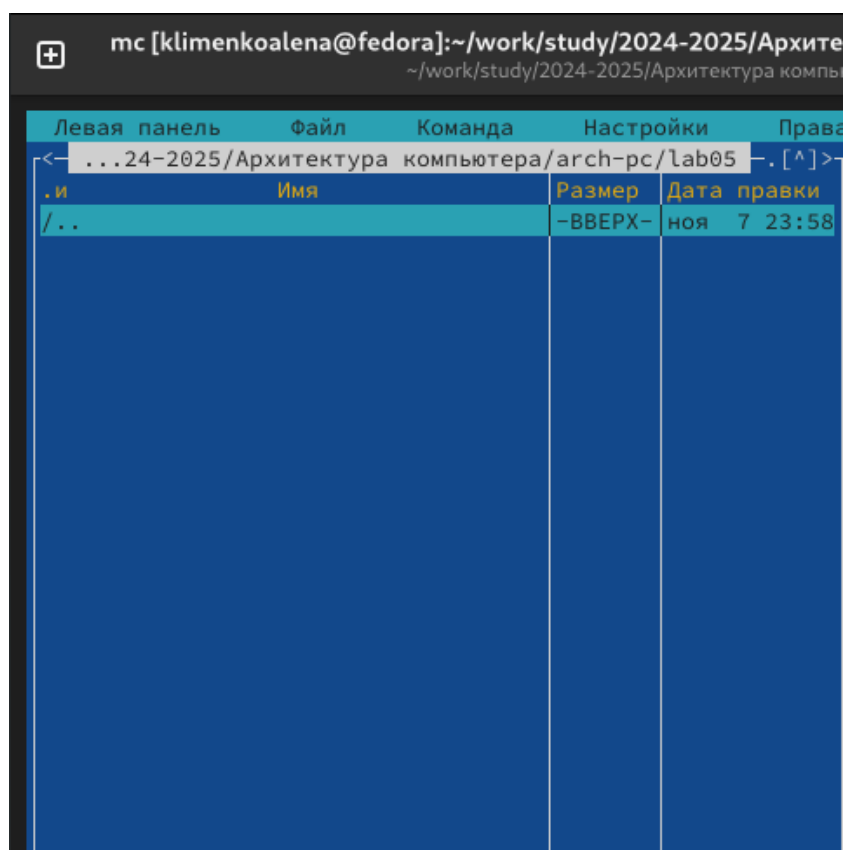


Рис. 4.4: Перемещение между директориями

В строке ввода прописываю команду `touch lab5-1.asm`, чтобы создать файл, в котором буду работать (рис. 4.5).

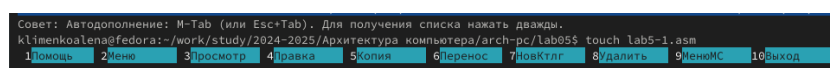


Рис. 4.5: Создание файла

4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano (рис. 4.6).

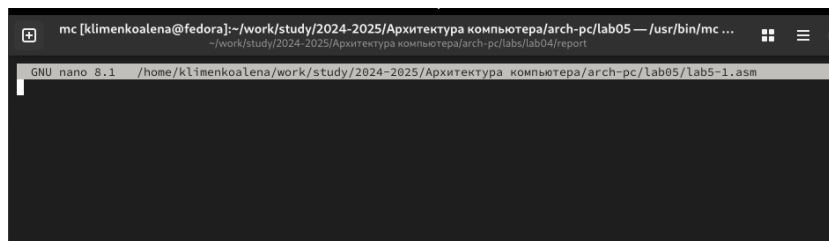


Рис. 4.6: Открытие файла для редактирования

Ввожу в файл код программы для запроса строки у пользователя (рис. 4.7).
Далее выхожу из файла, сохраняя изменения

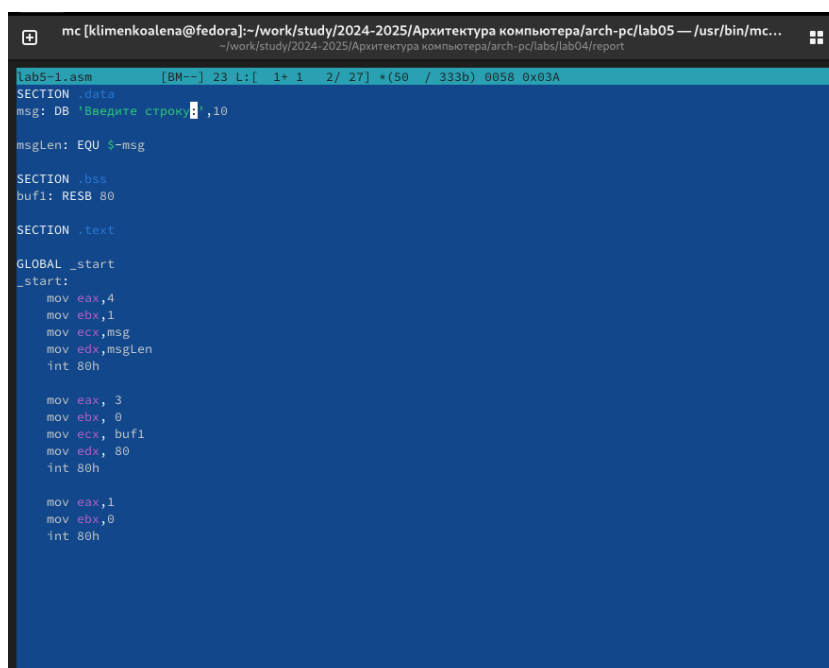


Рис. 4.7: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. 4.8).

```
mc [klimenkoalena@fedora]:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05 — /usr/bin/mc...
~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04/report

/home/klimenkoalena/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05/lab5-1.asm 333/333
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text

GLOBAL _start
_start:
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msgLen
    int 80h

    mov eax,3
    mov ebx,0
    mov ecx,buf1
    mov edx,80
    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Рис. 4.8: Открытие файла для просмотра

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-1.asm`. Создался объектный файл `lab5-1.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o` (рис. 4.9). Создался исполняемый файл `lab5-1`.

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-1
Введите строку:
Клименко Алёна Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$
```

Рис. 4.9: Компиляция файла и передача на обработку компоновщику

Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 4.11).

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-1
Введите строку:
Клименко Алёна Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$
```

Рис. 4.10: Исполнение файла

4.3 Подключение внешнего файла

Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” (рис. 4.11).



.и	Имя	Размер	Дата правки
/..	-ВВЕРХ-		ноя 7 23:53
/pandoc-3.5		16	окт 5 23:28
/pandoc-crossref-Linux		64	окт 10 04:19
in_out.asm		3942	ноя 8 00:34

Рис. 4.11: Скачанный файл

С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 4.12).

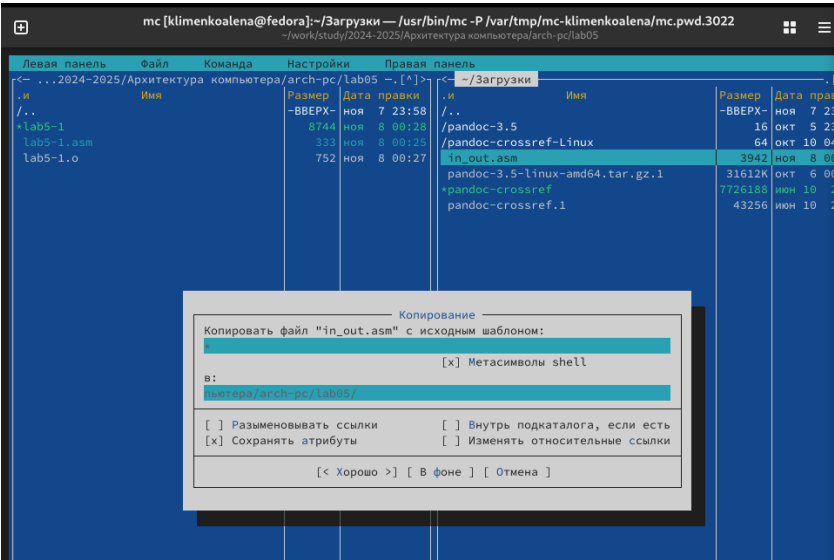


Рис. 4.12: Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла (рис. 4.13).

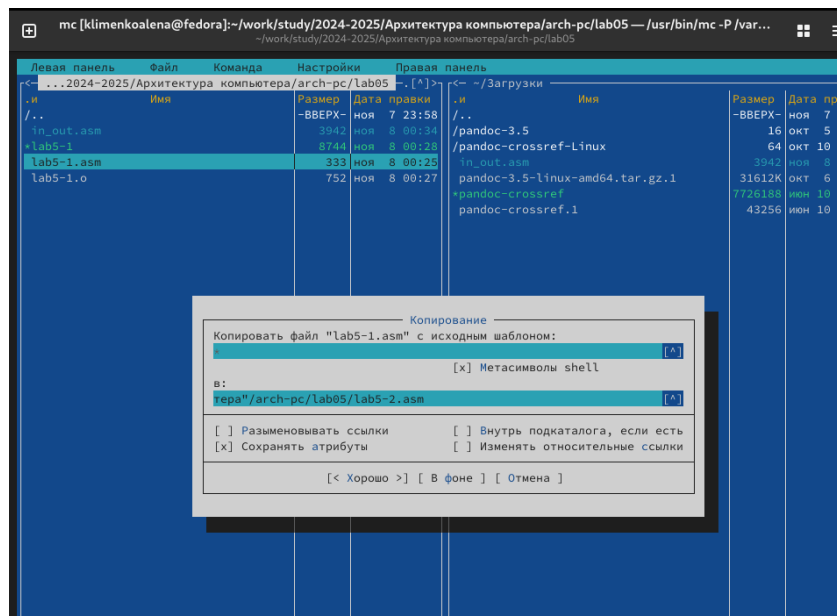


Рис. 4.13: Копирование файла

Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano (рис. 4.14), чтобы в программе использовались подпрограммы из внешнего файла in_out.asm.

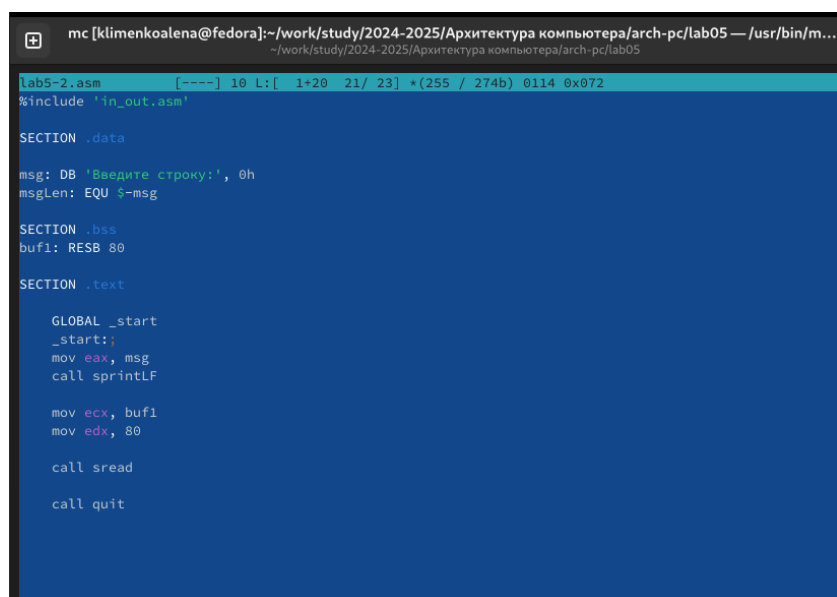


Рис. 4.14: Редактирование файла

Транслирую текст программы файла в объектный файл командой `nasm -f elf`

lab5-2.asm. Создался объектный файл lab5-2.o. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o` Создался исполняемый файл lab5-2. Запускаю исполняемый файл (рис. 4.15).

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку:
Клименко Алёна Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ mc

klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку:Клименко Алёна Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$
```

Рис. 4.15: Исполнение файла

Изменяю в файле подпрограмму `sprintLF` на `sprint` и снова транслирую файл выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл. (рис. 4.16).

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку:
Клименко Алёна Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ mc

klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку:Клименко Алёна Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$
```

Рис. 4.16: Отредактированный файл

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами `sprintLF` и `sprint`. ## Выполнение заданий для самостоятельной работы

1. Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 4.17).

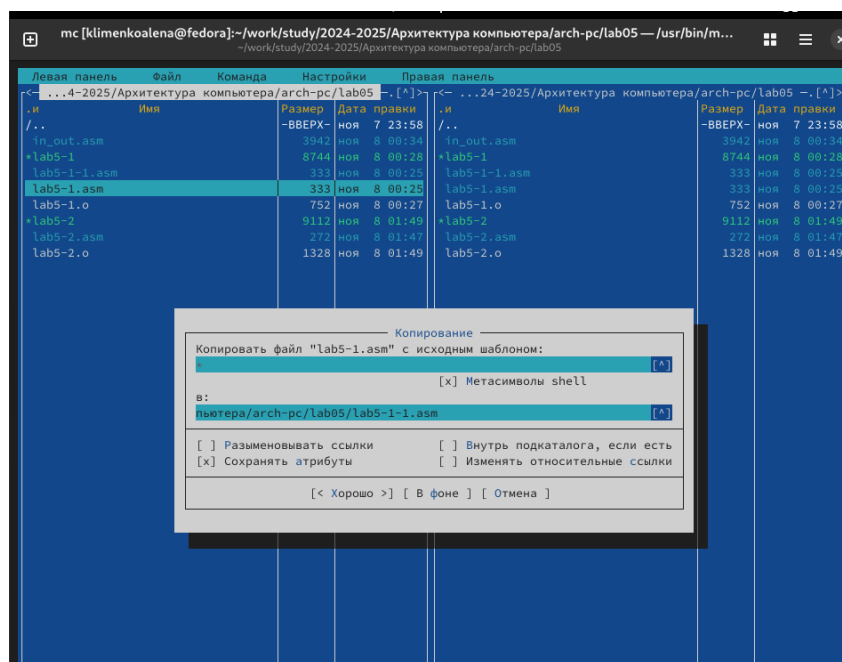


Рис. 4.17: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 4.18).

```
mc [klimenkoalena@fedora]:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05 — /usr/bin/m...
~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05
lab5-1-1.asm [-----] 0 L: [ 1+23 24/ 33] *(293 / 408b) 0010 0x00A
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msgLen
    int 80h

    mov eax,3
    mov ebx,0
    mov ecx,buf1
    mov edx,80
    int 80h

    mov eax,4
    mov ebx,1
    mov ecx,buf1
    mov edx,buf1
    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Рис. 4.18: Редактирование файла

2. Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. ??).

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Клименко Алена Сергеевна
Клименко Алена Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$
```

3. Создаю копию файла lab5-2.asm с именем lab6-2-1.asm с помощью функциональной клавиши F5 (рис. 4.19).

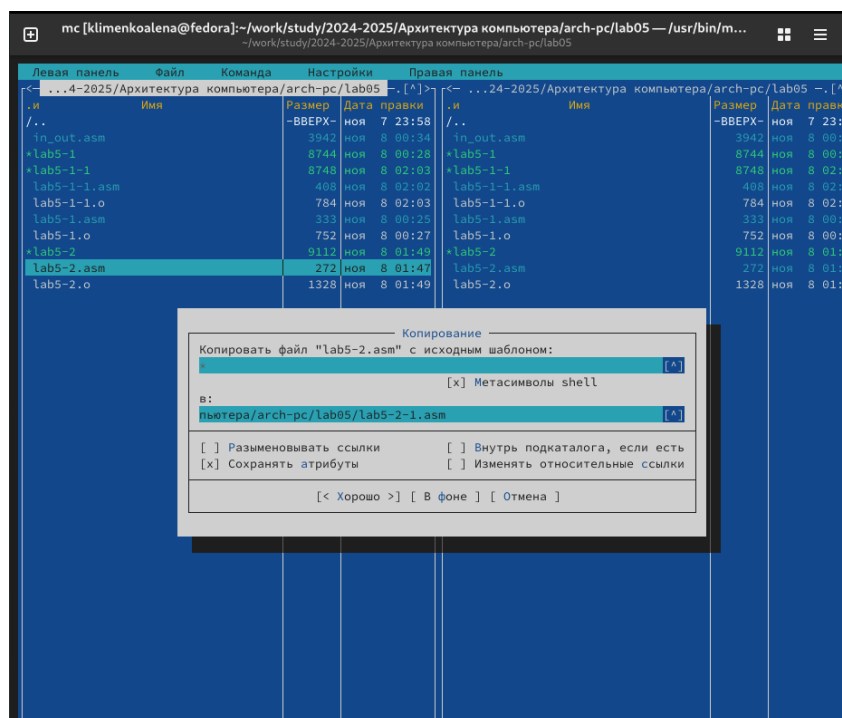


Рис. 4.19: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 4.20).

```
mc [klimenkoalena@fedora]:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05 — /usr/bin/m...
lab5-2-1.asm [----] 11 L: [ 1+24 25/ 27] *(313 / 328b) 0010 0x00A
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку:', 0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text

GLOBAL _start
_start:
mov eax, msg
call sprint

mov ecx, buf1
mov edx, 80

call sread
mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h

call quit
```

Рис. 4.20: Редактирование файла

4. Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 4.21).

```
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-2-1
Введите строку: Клименко Алёна Сергеевна
Клименко Алёна Сергеевна
klimenkoalena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$
```

Рис. 4.21: Исполнение файла

5 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера `mov` и `int`.

6 Список литературы

1. Лабораторная работа №5